

## PATENT

Atty. Dkt. No. NVDA/P000720

**IN THE CLAIMS:**

Please cancel claims 3, 5-10, 13-14, 17 and 19-20, and amend the claims as follows:

Claim 1 (Currently Amended) A method for caching versions of [[data]] constants for use by a plurality of execution units executing threads, comprising:

storing a first version of [[data]] a constant in a first level 1 cache together with a version tag for use by a first one of the execution units;

storing a second version of [[data]] a constant in a second level 1 cache together with a version tag for use by a second one of the execution units; [[and]]

storing one of the first and second versions of [[data]] a constant together with an updated version tag in a level 2 cache when one of the execution units requests a constant not available in one of the level 1 caches; and

updating the version tag with the constant in the level 2 cache to indicate the constant is associated with the oldest active thread.

Claim 2 (Currently Amended) The method according to claim 1, further comprising invalidating the second level 1 cache when the ~~second~~ version of [[data]] a constant is ~~no longer being used by an execution unit~~ transferred to a level 2 cache.

Claim 3 (Canceled)

Claim 4 (Currently Amended) The method according to claim 1, further comprising copying [[data]] a constant from a location in the level 2 cache to a location in a level 2 cache backup when the constant stored in the level 2 cache is not in use by any execution unit.

Claim 5-10 (Canceled)

Claim 11 (Currently Amended) A streaming processing array, comprising:

Page 2

431239\_1

PATENT

Atty. Dkt. No. NVDA/P000720

a first execution unit configured to process data and including a first level 1 cache adapted to store a constant and an associated version tag for use by the first execution unit;

a second execution unit configured to process data and including a second level 1 cache and adapted to store a constant and an associated version tag for use by the second execution unit; and

a level 2 cache coupled to both the first execution unit and the second execution unit for storing the constants transferred from the first and second level 1 caches when one of those caches must respond to a request for a constant by an associated execution unit, the version tag being updated by a controller configured to associate and update a version tag with the constant stored in each location in the level 2 cache, the constants stored in the level 2 cache being directly accessible by any one of the first and second execution units.

Claim 12 (Currently Amended) The streaming processing array of claim 11, further comprising a level 2 cache backup coupled to the level 2 cache, in the level 2 cache being configured to output a constant to the level 2 cache backup responsive to the controller.

Claim 13-14 (Canceled)

Claim 15 (Currently Amended) The streaming processing array of claim [[14]] 1, wherein the controller is configured to retire a location in the level 2 cache backup when a version tag associated with the [[data]] constant stored in the location is no longer being used by [[an]] one of the execution units.

Claim 16 (Currently Amended) The streaming processing array of claim 11, wherein the first level 1 cache is configured to output [[data]] constants to and to receive [[data]] constants from the second level 1 cache.

Claim 17 (Canceled)

Page 3

431239\_1

## PATENT

Atty. Dkt. No. NVDA/P000720

Claim 18 (Currently Amended) The streaming processing array of claim 11, wherein the streaming processing array resides within a programmable graphics processor coupled to a host computer, the processor being configured to control and output requests for constants to the execution units.

Claim 19-20 (Canceled)

Claim 21 (New) A method as claimed in claim 1, including locking the level 1 cache storing one of the constants until utilization of the locked constant is completed by the execution unit associated therewith.

Claim 22 (New) A method as claimed in claim 21, including moving the constant in the locked one of the level 1 caches to another level 1 cache, and unlocking the locked level 1 cache.

Claim 23 (New) A method as claimed in claim 1, including locking the level 1 cache storing one of the constants until the stored constant is moved to the level 2 cache.

Claim 24 (New) A method as claimed in claim 1, wherein each of the level 1 caches is associated with and accessed by a single one of the execution units, and the level 2 cache is associated with and directly accessed by a plurality of the execution units.

Claim 25 (New) The method according to claim 24, further comprising copying a constant from a location in the level 2 cache to a location in a level 2 cache backup when the constant stored in the level 2 cache is not in use by any execution unit.

Claim 26 (New) A method as claimed in claim 25, including locking the level 1 cache storing one of the constants until the stored constant is moved to the level 2 cache.

**PATENT**

Atty. Dkt. No. NVDA/P000720

Claim 27 (New) A method as claimed in claim 1, wherein a number of versions of constants stored is equal to the number of execution units.

Claim 28 (New) A streaming processing array as claimed in claim 11, the controller being adapted to lock the level 1 cache storing one of the constants until the stored constant is moved to the level 2 cache.

Claim 29 (New) A streaming processing array as claimed in claim 11, wherein each of the level 1 caches is associated with and accessed by a single one of the execution units and the level 2 cache is associated with and directly accessed by a plurality of the execution units.

Claim 30 (New) A streaming processing array as claimed in claim 11, the controller being adapted to move the constant in the locked on of the level 1 caches to another level 1 cache, and unlock the locked level 1 cache.

Claim 31 (New) A streaming processing array as claimed in claim 11, wherein a number of versions of constants stored is equal to the number of execution units.